Fast Face Verification Based on Gaussian Process Postprint

Authors: Zhou Siyang, Cao Lin

Date: 2018-05-20T00:00:00+00:00

Abstract

To address the issues of large training sample requirements, high computational complexity, and slow recognition speed in current face verification algorithms, a fast face verification method based on Gaussian processes in small-sample spaces is proposed. First, the conjugate gradient descent method is employed to learn the gradient directions of feature positions in key facial regions from training samples, thereby enabling feature localization for faces to be verified. Then, adaptive-scale local binary patterns are utilized to extract features, reducing feature dimensionality. Finally, the spectral mixture kernel function is adopted as the kernel function of the Gaussian process to classify the input facial features. Experiments conducted on the LFW, FERET, and Multi-PIE face databases demonstrate that the use of adaptive-scale local binary patterns effectively reduces feature dimensionality, and the combination of the Gaussian process model with the spectral mixture kernel can substantially decrease training samples while significantly improving both training and testing speeds.

Full Text

Fast Face Verification Based on Gaussian Processes

Zhou Siyang, Cao Lin†
Department of Communication Engineering, Beijing Information Science & Technology University, Beijing 100101, China

Abstract

To address the issues of large training sample requirements, high computational complexity, and slow recognition speed in current face verification algorithms, this paper proposes a fast face verification method based on Gaussian processes in small-sample spaces. First, the conjugate gradient descent method is used to learn the gradient directions of facial keypoint locations from training samples,

enabling feature localization for faces to be verified. Then, an adaptive multi-scale local binary pattern is employed for feature extraction to reduce feature dimensionality. Finally, a spectral mixture kernel function is adopted as the kernel of the Gaussian process to classify the input facial features. Experiments conducted on the LFW, FERET, and Multi-PIE face databases demonstrate that the adaptive multi-scale local binary pattern effectively reduces feature dimensions, while the combination of Gaussian process models with spectral mixture kernels significantly decreases training sample requirements and substantially improves both training and testing speeds.

Keywords: face verification; adaptive multi-scale local binary pattern; Gaussian process; spectral mixture kernel

0 Introduction

Face verification represents an important branch of face recognition with significant application value in surveillance, access control, image retrieval, automatic login, and automatic payment systems, making it a persistent research focus in computer vision for decades [1-3]. Despite its challenging nature, recent advances in deep learning and its application to face recognition [4-6] have improved verification accuracy. However, deep learning-based face verification requires training samples numbering in the hundreds of millions, demands extensive training time and high-performance hardware, and exhibits strong framework dependencies that hinder portability. Consequently, training face models in small-sample spaces remains a critical research area. Inspired by the cross-angle face recognition method proposed in literature [7] that achieved promising results, this paper introduces a fast face recognition approach based on Gaussian processes.

Feature extraction constitutes a crucial step in face verification. Traditional methods employ scanning windows that produce excessively high-dimensional features, leading to prohibitive computational costs. Considering the non-uniform distribution of facial information—where facial features contain richer information than other regions, with eyes containing the most information—this paper proposes an adaptive multi-scale feature extraction method (AMLBP). Experimental results demonstrate that this approach significantly reduces feature dimensions.

The classification model represents another key component in face verification, determining whether two input features belong to the same subject. While existing classification models [8-10] perform well on relatively simple face data distributions, their effectiveness diminishes substantially when dealing with complex distributions. This paper utilizes Gaussian processes as the classification model. As a non-parametric machine learning method, Gaussian processes can flexibly adapt to complex data distributions in real-world environments without requiring heuristic methods or manual parameter tuning. Their closed-form

marginal probability calculations enhance computational efficiency, and their hyperparameters can be learned automatically from training data without cross-validation for model selection, thereby reducing computational overhead.

In Gaussian process theory, the kernel function plays a decisive role in the model, with training samples used to train this kernel. Most existing literature employs traditional radial basis kernel functions [11], resulting in high computational costs and poor scalability. To address this, this paper replaces the conventional Gaussian process kernel with a spectral mixture kernel. Formed by modeling the kernel' s spectral density as a Gaussian mixture of scale locations, this kernel exhibits stronger extrapolation capabilities compared to radial basis kernels and can adapt to more complex data distributions.

Targeting the main challenges in current face verification, this paper introduces innovations and improvements in both feature extraction and classification stages, significantly reducing required training samples while enhancing training and testing speeds. Testing on the LFW database using the trained model achieved an accuracy of 98.7%.

1 Feature Extraction

Facial feature extraction must satisfy two fundamental requirements: first, identifying optimal features that maximize inter-person discrimination, and second, minimizing feature dimensionality to improve training and testing speeds. Mainstream algorithms include SIFT, LBP, HOG, and Gabor. While theoretically higher-dimensional features yield greater accuracy, literature [12] indicates that multi-scale feature extraction can improve algorithmic accuracy but significantly increases computational overhead, hindering practical implementation. This paper proposes an adaptive multi-scale feature extraction method that effectively reduces feature dimensions.

1.1 Feature Localization

Feature localization, the first step in feature extraction, specifies where features should be extracted. This paper employs a labeled face database [13] as training samples to learn the gradient directions at various landmark positions, which then determine feature extraction locations.

1.1.1 Principle Analysis Given an image I with n pixels containing m manual landmarks (Figure 1(a)), let ϕ denote a feature extractor (specifically AMLBP in this work). During training, faces with landmarks from the database are used as training samples, denoted as $\{x_i\}$. When a face is detected, an initial landmark $\bar{x}_0$, called the mean landmark (Figure 1(b)), is provided. Feature localization is achieved by minimizing Equation (1):

$$f(x) = \phi(I, x)$$

where x represents the landmark location. The training process starts from $\bar{x}_0$ and converges to x^*. To apply gradient-based methods, Equation (1) is Taylor-expanded around $\bar{x}_0$:

$$f(\bar{x}_0 + \Delta x) \approx f(\bar{x}_0) + J_f(\bar{x}_0)\Delta x + \frac{1}{2}\Delta x^T H_f(\bar{x}_0)\Delta x$$

where $J_f(\bar{x}_0)$ and $H_f(\bar{x}_0)$ are the Jacobian and Hessian matrices, respectively. Taking the partial derivative of Equation (2) with respect to Δx and setting it to zero yields the first update step for Δx:

$$\Delta x_1 = -H^{-1}J^T = -2H^{-1}R^T(\phi(\bar{x}_0) - \phi(x^*))$$

In the first gradient iteration, this is treated as $\Delta x_1 = -2H^{-1}R^T h$, where $h = \phi(\bar{x}_0) - \phi(x^*)$ and $R = J_f(\bar{x}_0)$. To avoid computing the Hessian and Jacobian matrices, h is treated as the gradient direction. By learning a linear regression between Δx and h, R can be obtained directly, simplifying Equation (3) to:

$$\Delta x_1 = R_1 h + b_1$$

where b_1 is a bias term. For a specific image, Newton' s method is applied along the gradient direction to determine new iterations using the previously learned R and b.

1.1.2 Learning Gradient Directions This section describes how to learn R and b from training data. Let the i-th face image be $\{I_i\}$ with corresponding landmarks $\{x_i^*\}$. At each update step, Equation (6) uses the previously learned R_{k-1} and b_{k-1} to generate a new data pair $\{x_{k-1}, \Delta x_k\}$. The parameters R_k and b_k are learned by minimizing:

$$\arg\min_{R_k, b_k} \sum_i ||\Delta x_{ik} - R_k h_i(x_{ik}) - b_k||^2$$

This unconstrained optimization problem is effectively solved using the conjugate gradient method, which requires only first-order derivative information for rapid convergence. Applying conjugate gradient descent to minimize the objective function (Equation (7)) yields the gradient directions for feature locations. When a face is detected, landmarks can be localized based on these gradient directions, with localization results shown in Figure 2.

1.2 Adaptive Scale Feature Extraction

Considering that different facial regions contain varying amounts of information—for instance, eyes contain the most sensitive and discriminative features for face recognition—the feature extraction scale should be largest in such regions. Conversely, cheeks contribute less to face recognition and should use smaller extraction scales, thereby reducing feature dimensions without information loss. Accordingly, this paper proposes an adaptive multi-scale feature extraction method that adjusts extraction scales based on feature sensitivity.

To obtain facial feature sensitivity, this paper maps face features trained by Adaboost [12] to those acquired in Section 1.1, then determines extraction scales based on each weak classifier' s error rate (treated as sensitivity). Experimental results (Figure 3) indicate that a feature extraction scale of 4 achieves optimal classification performance. Therefore, the maximum feature extraction scale is set to 4, and features are categorized into four classes $\{k_1, k_2, k_3, k_4\}$. The Adaboost algorithm trained 200 facial features, sorted by error rate from low to high. Features obtained in Section 1.2 are classified into four categories using Euclidean distance: for each feature to be classified, the nearest Adaboost feature' s category is assigned. Equations (8) and (9) detail the extraction method, where $p(\cdot)$ denotes feature location, $D(\cdot)$ represents Euclidean distance, $K(\cdot)$ is the scale corresponding to $p(\cdot)$, x_{th} is the feature location obtained via the method in Section 2.1, and x_{Ad} is the Adaboost feature location.

Multi-scale features are extracted using a Gaussian pyramid:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

where $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$ and $\sigma_i = s_i \cdot 2^{k/2}$ for $i = 1, 2, 3, 4$ and $k = 0, 1, 2, ...$ The Gaussian scale space $L(x, y, \sigma)$ enables multi-scale feature extraction, as shown in Figure 4. LBP features are then extracted from each feature' s Gaussian scale space to obtain adaptive multi-scale features.

Figure 5 compares classification performance using adaptive scales versus single scales. Analysis reveals that while accuracy increases with feature extraction scale, so does dimensionality and computational cost. However, adaptive scale classification achieves accuracy comparable to $k = 5$ while using an average scale of 2.6, significantly reducing feature dimensions. The adaptive scale mapping is shown in Table 1, where feature indices correspond to those in Figure 6.

2 Gaussian Process Model

Most current face verification classification models are parametric [14-15], requiring pre-defined parameters that directly affect classification performance. Different face data distributions necessitate different parameter settings, resulting in poor performance when trained models are applied to natural scenes [16]. Gaussian processes, as non-parametric stochastic processes, can adapt to

complex data distributions in real-world scenarios without heuristic methods or manual parameter tuning. As a Bayesian kernel method, Gaussian processes offer efficient closed-form marginal probability calculations, overcome overfitting, and exhibit strong robustness. Moreover, hyperparameters are learned automatically from training data. Leveraging these advantages, this paper adopts Gaussian processes as the classifier.

2.1 Gaussian Process as Binary Classifier

Consider a training set with N samples: $\{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$ and $y_i = \pm 1$. The joint probability of training data is calculated using Equation (11). For test data x^* with output y^*, the predictive distribution for any element in the test set is a Gaussian joint distribution over n training samples and 1 test sample. The core idea of Gaussian classification involves setting a latent function $f(x)$ and using a logistic function $\pi(\cdot)$ to incorporate prior knowledge, outputting class probabilities in the interval $[0, 1]$. Prediction involves two steps: first, computing the latent variable distribution for the test set (Equation 14); then, calculating the probability prediction using the latent variable distribution (Equation 15).

Treating Gaussian processes as binary classifiers results in a non-Gaussian likelihood function $p(y|f)$, causing the posterior $p(f|X, y)$ to also be non-Gaussian and preventing analytical computation of $p(y^*|X, y, x^*)$. Therefore, the Laplace approximation is applied to approximate the posterior $p(f|X, y)$ as Gaussian, yielding:

$$q(f|X, y) = \mathcal{N}(f|\hat{f}, (K^{-1} + W)^{-1})$$

where $\hat{f} = \arg\max_f p(f|X, y)$ and $W = -\nabla\nabla \log p(y|f)|_{f=\hat{f}}$.

2.1.1 Gaussian Process Kernel Function As a kernel method, Gaussian process performance is entirely determined by its kernel function. Current research focuses on kernels such as the squared exponential kernel, Gibbs kernel, and deep kernel [17]. However, these methods only suit small training samples, have limited extrapolation ability, and incur high computational costs: $\mathcal{O}(n^3)$ for training, $\mathcal{O}(n^2)$ for storage, and $\mathcal{O}(n^2)$ for prediction. To address these limitations, this paper employs a spectral mixture kernel for face verification with Gaussian processes. Experiments demonstrate that this kernel effectively discovers face patterns with strong extrapolation capabilities, performing well in natural scene face verification. Its simple closed-form enables direct analysis and inference, with the simplified form:

$$k_{SM}(x, x'|\theta) = \sum_{q=1}^{Q} w_q \prod_{d=1}^{D} \exp(-2\pi^2 \sigma_{q,d}^2 (x_d - x'_d)^2) \cos(2\pi \mu_{q,d} (x_d - x'_d))$$

Literature [18] indicates this kernel encompasses many stationary kernels and can approximate them with arbitrary precision. Training on the Multi-PIE database and comparing with traditional kernels (Figure 7) shows that the spectral mixture kernel achieves higher accuracy at the same false alarm rate, validating its effectiveness.

3 Experiments and Analysis

The experimental platform uses an Intel Core i5 processor at 2.70GHz with 12GB memory. C++ programming with OpenCV 2.4.9 is used for image preprocessing. Three face databases are employed: LFW, FERET, and Multi-PIE (partial). LFW contains 13,233 faces from 5,750 individuals in unconstrained settings, encompassing variations in pose, expression, age, gender, hairstyle, and other parameters, making it the most challenging standard face database in academia. It is used for testing in this paper. FERET comprises 21,833 grayscale faces of 1,040 individuals under the same scene with different lighting conditions, including various poses and expressions, making it one of the most widely used face databases. Multi-PIE contains 750,000 faces of 337 individuals under different poses, illuminations, and expressions in controlled environments. Since the complete database is not publicly available, 20,000 faces are used for model training.

3.1 Experimental Procedure

The face verification process is illustrated in Figure 8: (a) Learn gradient directions for each feature from the template database using the method in Section 2.1 and apply them for feature localization; (b) Extract features at localized positions using the proposed adaptive scale method; (c) Compute feature similarity using Euclidean distance; (d) Input similarity measures into the Gaussian model to obtain classification results.

3.2 Experimental Results Analysis

This paper employs adaptive scales for feature extraction to achieve dimensionality reduction, uses Gaussian processes as binary classifiers to significantly reduce training samples, and replaces the squared exponential kernel with a spectral mixture kernel to enhance extrapolation capability. Each component is analyzed below.

3.2.1 Adaptive Scale Dimensionality Reduction Analysis Adaptive scale extraction is compared with LE, LBP, SIFT, HOG, and Gabor algorithms in terms of dimensionality and accuracy (Table 2). The adaptive scale algorithm achieves a feature dimension of 1.5×10^5—equivalent to 1/4 of LE' s dimension but with 6.2% higher accuracy, and 3/4 of Gabor' s dimension with 7.7% higher accuracy. These results confirm that the adaptive scale method significantly reduces feature dimensions without compromising accuracy.

3.2.2 Gaussian Process Model Performance Analysis The Gaussian process model is compared with existing binary classifiers: SVM, Logistic Regression (LR), and Adaboost (Table 3). Analysis shows that Gaussian processes maintain accuracy above 98% across different training sets, while other classifiers exhibit lower overall accuracy and greater fluctuations when trained on different datasets. This demonstrates that Gaussian process models can adapt to various data distributions without dataset-specific parameter tuning.

3.2.3 Superiority of Spectral Mixture Kernel Function Replacing the squared exponential kernel with a spectral mixture kernel enhances the Gaussian process classifier' s extrapolation ability. To validate this advantage, models are trained on constrained-scenario faces from FERET and Multi-PIE and tested on unconstrained LFW faces, then compared with the squared exponential kernel. Figures 9 and 10 show that the spectral mixture kernel achieves significantly higher accuracy, demonstrating superior extrapolation performance—models trained on constrained data perform well on natural-scenario test sets. Additionally, Figure 11 presents results when LFW data is split into training and testing sets, again showing the spectral mixture kernel' s significant advantage. Figure 12 illustrates the spectral mixture kernel' s performance across different training and test data, confirming that models trained on either natural or constrained scenarios achieve good test results.

Furthermore, Gaussian process models substantially reduce training samples compared to mainstream deep learning face recognition algorithms (Table 4). With only 20K training samples, GP achieves 98.7% accuracy—matching FaceNet' s accuracy but using approximately 25× fewer data (FaceNet uses 494K). Compared to DeepID, OpenFace, and DeepFace, GP achieves slightly higher accuracy while dramatically reducing training samples.

4 Conclusion

Experimental validation and analysis lead to the following conclusions: (a) The proposed adaptive multi-scale algorithm extracts facial features adaptively according to the non-uniform distribution of facial information, significantly reducing dimensions compared to traditional methods; (b) The Gaussian process model leverages its non-parametric stochastic process characteristics, automatically adapting to different training data without parameter modifications, thereby reducing training samples while enhancing system robustness; (c) The spectral mixture kernel improves extrapolation performance over traditional radial basis functions, enabling models trained on constrained-scenario faces to achieve high accuracy on natural-scenario test sets.

The proposed face verification algorithm significantly improves training speed and testing speed while requiring fewer training samples and no platform/framework dependencies, facilitating deployment and migration.

References

[1] Kumar N, Berg A C, Belhumeur P N, et al. Attribute and simile classifiers for face verification [C]// Proc of IEEE International Conference on Computer Vision. 2010: 365-372.
[2] Huang G B. Learning hierarchical representations for face verification with convolutional deep belief networks [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2012: 2518-2525.
[3] Milanfar H J S P. Face Verification Using the LARK Face, Representation [J]. IEEE Trans on Information Forensics & Security, 2011, 6 (4): 1275-1286.
[4] Sun Y, Wang X, Tang X. Deep Learning Face Representation from Predicting 10,000 Classes [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2014: 1891-1898.
[5] Sun Y, Liang D, Wang X, et al. DeepID3: face recognition with very deep neural networks [J]. Computer Science, 2015.
[6] Sun Y, Wang X, Tang X. Deep Learning Face Representation by Joint Identification-Verification [J]. 2014, 27: 1988-1996.
[7] Ben X, Meng W, Yan R, et al. An improved biometrics technique based on metric learning approach [J]. Neurocomputing, 2012, 97 (1): 44-51.
[8] Zhu Z, Luo P, Wang X, et al. Deep learning identity-preserving face space [C]// Proc of IEEE International Conference on Computer Vision. 2013: 113-
[9] Sun Y, Wang X, Tang X. Hybrid deep learning for face verification [C]// Proc of IEEE International Conference on Computer Vision. 2014: 1489-
[10] Chen D, Cao X, Wang L, et al. Bayesian face revisited: a joint formulation [C]// Proc of European Conference on Computer Vision. Springer-Verlag, 2012: 566-579.
[11] Kim S J, Magnani A, Boyd S. Optimal kernel selection in Kernel Fisher discriminant analysis [C]// Proc of International Conference on Machine Learning. 2006: 465-472.
[12] Chen D, Cao X, Wen F, et al. Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2013: 3025-3032.
[13] Sagonas C, Tzimiropoulos G, Zafeiriou S, et al. 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge [C]// Proc of IEEE International Conference on Computer Vision Workshops. Washington DC: IEEE Computer Society, 2013: 397-403.
[14] Jones M J, Viola P. Robust Real-time Object Detection [J]. International Journal of Computer Vision, 2001, 57 (2): 87.
[15] Zhu Z, Luo P, Wang X, et al. Deep Learning Identity-Preserving Face Space [C]// Proc of IEEE International Conference on Computer Vision. 2013: 113-
[16] Wright J, Hua G. Implicit elastic matching with random projections for pose-variant face recognition [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2009: 1502-1509.
[17] Wilson A G, Hu Z, Salakhutdinov R, et al. Deep kernel learning [J]. Com-

puter Science, 2015.
[18] Wilson A G. Covariance kernels for fast automatic pattern discovery and extrapolation with Gaussian processes [J]. University of Cambridge, 2014.
[19] Schroff F, Kalenichenko D, Philbin J. FaceNet: A unified embedding for face recognition and clustering [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC: IEEE Computer Society, 2015: 815-823.
[20] Amos B, Ludwiczuk B, Satyanarayanan M. Openface: A general-purpose face recognition library with mobile applications [J]. CMU School of Computer Science, 2016.
[21] Taigman Y, Yang M, Ranzato M, et al. DeepFace: Closing the Gap to Human-Level Performance in Face Verification [C]// Computer Vision and Pattern Recognition. 2014: 1701-1708.
[22] Ng H W, Winkler S. A data-driven approach to cleaning large face datasets [C]// Proc of IEEE International Conference on Image Processing. 2015:

Note: Figure translations are in progress. See original paper for figures.